SCHEDULING MEANS FOR DATA SWITCHING APPARATUS

This invention relates to scheduling means for data switching apparatus for use in computer-controlled digital data switching systems.

Many types of data transmission apparatus are known, all having their own particular features and systems. In all cases the intention is to allow data switching and transmission to be achieved as rapidly as the apparatus will allow. It is common for data to be sent in "packets" consisting of a predetermined number of bits of data plus control information indicating certain parameters of the data or its mode of transmission.

In data switching apparatus having an number of users there may be a requirement at any one time to set up a number of different interconnections between input ports and output ports. In any form of switch there is a limit to the number of simultaneous interconnections that may be formed. The switch is operating at its greatest efficiency when the greatest possible number of interconnections is formed and switching apparatus frequently includes what may be termed "scheduling means" in order to achieve this maximum number of interconnections.

A good scheduling scheme needs to balance the potentially conflicting objectives of making sure that all output ports are connected where there are requests for a connection to that port (efficiency), that high priority traffic is serviced quickly (prioritisation) and that low priority traffic is not ignored (fairness). The present invention addresses all of these issues and may, for example, be used with the data switching apparatus described and claimed in our co-pending British patent application No. 9717412.2

Various types of scheduling means are known. For example, United States Patent No. 5,500,858 describes one form of scheduling means in which requests for interconnections are considered and satisfied using what are called "rotating priority iterative matching desynchronising scheduler units". The "priority" in this case refers to priority given to input and output ports at any given time so as to ensure that each port has a fair chance of having a connection requests satisfied. The U.S. Patent goes on to describe how the scheme could be extended to handle requests at multiple priority levels but the scheme described would lack fairness, that is low priority requests would be ignored

under heavy load conditions where only higher priority requests would be satisfied.

It is an object of the present invention to provide data switching apparatus which includes scheduling means operable to satisfy a greater number of requests for interconnections than has previously been possible under such circumstances.

According to the present invention there is provided scheduling means for data switching apparatus having a plurality of input ports and a plurality of output ports between which data having one of a predetermined number of priority levels is to be passed, which scheduling means includes a first pipeline stage operable to satisfy at least some of the requests for interconnections which are applied to the scheduling means, a priority mixer to which are applied those requests for interconnections which were not satisfied by the first pipeline stage together with requests of different priority levels and operable to select which of those requests should be further considered, and at least one further pipeline stage to which are applied said further requests and operable to satisfy such of those requests as are possible and were not satisfied by any preceding pipeline stage.

The present invention overcomes the problems associated with the known prior art by using existing types of scheduling units, (for example those described in U.S. Patents Nos. 5,500,858 and 5.267,235, though any scheduling means which operates as described herein may be used) but connecting them in a novel arrangement. The scheduling means to be described may, for example, be used with the data switching apparatus described and claimed in our co-pending British patent application No. 9717412.2.

The invention will now be described with reference to the accompanying drawing, which shows a schematic block diagram of one embodiment of the invention.

The drawing shows three pipeline stages 10 to 12, with a priority mixer 13 connected between pipeline stages 10 and 11. Input connections and output connections are provided to the various pipeline stages and the priority mixer as shown and the operation of the arrangement will be described below.

Each of the pipeline stages 10, 11 and 12 operates to receive input connection Request Vectors RVi at Priority level Pi and a

Connection Vector CVi. In response to these inputs the pipeline stage generates output signals Queue Return QRet, Request Vector out RVo, Priority out Po and Connection Vector out CVo. The Request Vectors are bit fields where each bit corresponds to a possible connection between one of the input ports and one of the output ports of the data switching apparatus. That is, if there are n input ports and m output ports, the Request Vectors will be n x m bits wide, where a bit that is set indicates that a connection is being requested from the corresponding input port to the corresponding output port, whilst a bit that is clear indicates that such a connection is not being requested at this time. The Priority fields Pi and Po indicate the priority of the connection being requested at input (RVi) and output (RVo) respectively. The connection requests from each input port are all of the same priority, though the connections requested from different input ports may be of different priorities. The Connection Vector signal CVo defines connections which are to be made by a switching matrix (not shown). They indicate which input port, if any, is to be connected to each output port of the data switching apparatus. The Queue Return signals Qret represent connection requests that cannot be satisfied. These requests are returned to the input queues of the data switching apparatus ready to be requested again. The operation performed by each pipeline stage is to consider the connection requests at RVi and satisfy as many of them as possible, adding details of each satisfied connection to any already present at CVi and presenting the combined set of connections at CVo. Since each input port and each output port may only be involved in one connection at any given time, any connection requests which involve an input port or output port which is already part of a satisfied connection request may no longer be satisfied within the present set of connections and so such requests are returned to the input queues (signal Qret), to be considered as part of a subsequent set of connections. The remaining connection requests (those for which the corresponding input and output ports are still available for connection) are presented at the RVo output of the pipeline stage in order for them to be considered by a subsequent pipeline stage. Any such requests at the output of the last pipeline stage 13 (where there is no subsequent pipeline stage to consider them) are returned to the input queues of the data switching apparatus, as with the Qret output.

Consider now the overall operation of the scheduling means described above. At the input to the first pipeline stage, 10 requests for connections RVi at a single priority level Pi are presented. The first pipeline stage 10 then attempts to satisfy as many of these requests as possible.Traffic of each priority level is presented to the first pipeline stage 10 at a frequency proportional to the required bandwidth allocation for that priority level. For example, high priority level requests could be presented 50% of the time if a 50% bandwidth allocation for high priority traffic was required. The proportions assigned to each priority level would depend on the application and would be assigned by the system administrator and be independent of the operation of the pipeline stage. A lookup table may be used to define the priorities for each priority level. If there is only a small number of requests at the priority Pi then the first pipeline stage 10 will not make many connections and most of the input and output ports will not be utilised within the set of connections created by this stage, nor will there be many connection requests outstanding at that priority level which may be satisfied by the remaining pipeline stages 11 and 12. For this reason, the priority mixer 13 is introduced between the first and second pipeline stages 10 and 11. Applied to priority mixer 13 are connection requests RV2i of priorities other than Pi, the priorities of the requests being denoted by P2i. The priority mixer 13 decides, for each input port, whether to pass on to the second pipeline stage 11 the requests RVi remaining at priority level Pi from the first pipeline stage 10 or the new requests RV2i. The decision is made on the basis of choosing whichever set of requests has the highest number of requests that could still be satisfied within the current Connection Vector CVi, taking into account which input and output ports are already used by satisfied connection requests. This leads to higher connectivity within the data switching apparatus than if only requests of a single priority were considered, that is it is more efficient. It also allows good performance for low priority traffic in the absence of any higher priority traffic, since the low priority requests may be presented at the second pipeline stage 11 via the priority mixer 13, regardless of how infrequently low priority requests are selected to be presented to the first pipeline stage 10. The Connection Vector output CVo of the

last pipeline stage 13 defines all the connections that are to be made by the data switching apparatus in the next cycle of operation.

Since mixing requests of different priorities at the second and subsequent stages of the scheduling means leads to greater efficiency, it might seem like a good idea to do the same at the first pipeline stage 10. However, the scheduling units that make up each pipeline stage

do not themselves take any account of the priority of each request, that is they treat all requests equally. Hence if requests of different priorities were presented to the first pipeline stage there would be no concept of priority at all within the scheduling means, since the low priority requests would compete equally with the high priority requests for connections. Thus it will be seen that the first pipeline stage provides prioritisation and fairness as defined above, whilst the subsequent pipeline stages provide efficiency.

Further pipeline stages may be added if it is felt that three stages are not able to provide sufficiently high efficiency of switch utilisation. There is a trade-off between switch utilisation (how many connection requests may be satisfied at any time) and latency (each pipeline stage takes time to operate), so that the number of pipeline stages required will depend upon what balance of these factors is required for a particular embodiment. In general, more pipeline stages are needed to create maximal sets of connections as the number of ports in the data switching apparatus increases. In addition, priority mixer elements may be placed between others of the pipeline units to further increase efficiency if desired.

Instances may occur when, for example, certain connections between input ports and output ports are to be retained for a period of time (static or permanent connections). Alternatively, it may be necessary at certain times to block specified input ports or output ports (during a period of system maintenance, for example). Similarly, it may be necessary at certain times to set up connections from one input port to more than one output port at the same time (multicast). All of these facilities may be incorporated into data switching apparatus which uses the scheduling means described above. This is done by connecting appropriate logic to the inputs of the first pipeline stage 10 and/or the priority mixer 13. For instance, to create a permanent connection between an input port and an output port, the CVi input of the first pipeline stage 10 could be preset to indicate the required connection(s) rather than having all of its bits clear (indicating no pre-existing connections). Input and output ports may be blocked by masking off the appropriate bits of the RVi input of the first pipeline stage 10 and the RV2i input of the priority mixer 13. Multicast connections may be made in the same way as permanent connections, except that more than one

output port is set up to be connected to the desired input port. In all of these cases, the scheduling means works around the existing connections or blocked ports, making whatever connections it can between the remaining unconnected and non-blocked input and output ports.